

1 **What is claimed is:**

2 1. A system for performing kernel-mode operations
3 comprising:

4 a kernel-mode interface generator for generating a
5 kernel-mode interface driver, which in turn
6 generates a call gate to perform a kernel-mode
7 operation with kernel-mode authorization in a
8 kernel mode; and

9 an authorization interface, coupled to the kernel-
10 mode interface generator, to connect a user
11 mode to kernel mode, switching a process from
12 user mode to kernel mode via the call gate to
13 perform the kernel-mode operation.

1 2. The system as claimed in claim 1, wherein the
2 authorization interface sends a call gate request to the
3 kernel-mode interface generator to generate the kernel-
4 mode interface driver, the call gate generated
5 accordingly, and the authorization interface instructing
6 the process to enter the kernel mode through the call
7 gate.

1 3. The system as claimed in claim 1, wherein the
2 kernel-mode operation is an operation with Ring 0
3 authorization level in the kernel mode.

1 4. The system as claimed in claim 1, wherein the
2 process, a user-mode operation, is capable of user-mode
3 authorization in a protected mode.

1 5. The system as claimed in claim 4, wherein the
2 user-mode authorization is Ring 3 authorization level in
3 the protected mode.

1 6. The system as claimed in claim 4, wherein the
2 call gate sets a call gate selector and an entry point in
3 a global descriptor table, having a call gate descriptor
4 and a code-segment descriptor, to enable the process to
5 perform the operation with kernel-mode authorization in
6 the kernel mode.

1 7. The system as claimed in claim 6, wherein the
2 user-mode authorization of the process is switched to
3 kernel-mode authorization by the call gate selector via
4 the entry point in the global descriptor table, and is
5 switched back after the operation with kernel-mode
6 authorization has been performed.

1 8. The system as claimed in claim 7, wherein a far
2 call stated by the call gate selector points to the call
3 gate descriptor, and a CPU switches an instruction
4 pointer to the entry point, when a caller from the call
5 gate gives a call, if the caller has kernel-mode
6 authorization.

1 9. The system as claimed in claim 8, wherein the
2 instruction pointer has kernel-mode authorization, is
3 switched to the entry point, to perform the operation
4 with kernel-mode authorization in the kernel mode, and is
5 switched back to the user-mode authorization after the

6 operation with kernel-mode authorization has been
7 performed.

1 10. A method for performing kernel-mode operations
2 comprising steps of:

3 providing a kernel-mode generator;
4 generating a kernel-mode interface using the kernel-
5 mode generator to generate a call gate
6 performing a kernel-mode operation with kernel-
7 mode authorization in a kernel mode;
8 providing an authorization interface to connect a
9 user mode to the kernel mode; and
10 switching a process from the user mode to the kernel
11 mode via the call gate through the
12 authorization interface to perform the kernel-
13 mode operation with kernel-mode authorization.

1 11. The method as claimed in claim 10, further
2 comprising, in the step of providing the authorization
3 interface:

4 sending of a call gate request by the authorization
5 interface to the kernel-mode interface
6 generator to generate the kernel-mode interface
7 driver;
8 generating the call gate using the kernel-mode
9 interface driver; and
10 the authorization interface instructing the process
11 to enter the kernel mode through the call gate.

1 12. The method as claimed in claim 10, wherein in
2 the generating step, the kernel-mode operation is an

3 operation with Ring 0 authorization level in the kernel
4 mode.

1 13. The method as claimed in claim 10, wherein in
2 the switching step, the process, a user-mode operation,
3 is capable of user-mode authorization in a protected
4 mode.

1 14. The method as claimed in claim 13, wherein the
2 user-mode authorization is Ring 3 authorization level in
3 the protected mode.

1 15. The method as claimed in claim 13, wherein in
2 the generating step, the call gate sets a call gate
3 selector and an entry point in a global descriptor table,
4 having a call gate descriptor and a code-segment
5 descriptor, to perform the operation with kernel-mode
6 authorization in the kernel mode.

1 16. The method as claimed in claim 14, wherein the
2 user-mode authorization of the process is switched to
3 kernel-mode authorization by the call gate selector via
4 the entry point in the global descriptor table, and is
5 switched back after the operation with kernel-mode
6 authorization has been performed.

1 17. The method as claimed in claim 16, wherein a
2 far call stated by the call gate selector points to the
3 call gate descriptor, a CPU switches an instruction
4 pointer to the entry point, when a caller from the call
5 gate sends a call, if the caller has kernel-mode
6 authorization.

1 18. The method as claimed in claim 17, wherein the
2 instruction pointer has kernel-mode authorization, is
3 switched to the entry point, to perform the operation
4 with kernel-mode authorization in the kernel mode, and is
5 switched back to user-mode authorization after performing
6 the operation with kernel-mode authorization.

1 19. A storage medium for storing a computer program
2 providing a method for performing kernel-mode operations,
3 comprising using a computer to perform the steps of:

4 providing a kernel-mode generator;

5 generating a kernel-mode interface using the kernel-

6 mode generator to generate a call gate with

7 performing a kernel-mode operation with kernel-

8 mode authorization in a kernel mode;

9 providing an authorization interface to connect a

10 user mode to the kernel mode; and

11 switching a process from the user mode to the kernel

12 mode via the call gate through the

13 authorization interface to perform the kernel-

14 mode operation with kernel-mode authorization.

1 20. The storage medium as claimed in claim 19,
2 wherein the authorization interface sends a call gate
3 request to the kernel-mode interface generator to
4 generate the kernel-mode interface driver, the call gate
5 is generated according thereto, and the authorization
6 interface directs the process to enter the kernel mode
7 through the call gate.

1 21. The storage medium as claimed in claim 19,
2 wherein the kernel-mode operation is an operation with
3 Ring 0 authorization level in the kernel mode.

1 22. The storage medium as claimed in claim 19,
2 wherein the process, a user-mode operation, is capable of
3 user-mode authorization in a protected mode.

1 23. The storage medium as claimed in claim 22,
2 wherein the user-mode authorization is Ring 3
3 authorization level in the protected mode.

1 24. The storage medium as claimed in claim 23,
2 wherein the call gate sets a call gate selector and an
3 entry point in a global descriptor table, having a call
4 gate descriptor and a code-segment descriptor, to perform
5 the operation with kernel-mode authorization in the
6 kernel mode.

1 25. The storage medium as claimed in claim 24,
2 wherein the user-mode authorization of the process is
3 switched to kernel-mode authorization by the call gate
4 selector via the entry point in the global descriptor
5 table, and is switched back after the operation with
6 kernel-mode authorization has been performed.

1 26. The storage medium as claimed in claim 25,
2 wherein a far call stated by the call gate selector
3 points to the call gate descriptor, a CPU switches an
4 instruction pointer to the entry point, when a caller
5 from the call gate sends a call, if the caller has
6 kernel-mode authorization.

1 27. The storage medium as claimed in claim 26,
2 wherein the instruction pointer has kernel-mode
3 authorization, is switched to the entry point, to perform
4 the operation with kernel-mode authorization in the
5 kernel mode, and is switched back to user-mode
6 authorization after performing the operation with kernel-
7 mode authorization.